

ON-OFF (kétállású) hőmérsékletszabályozás

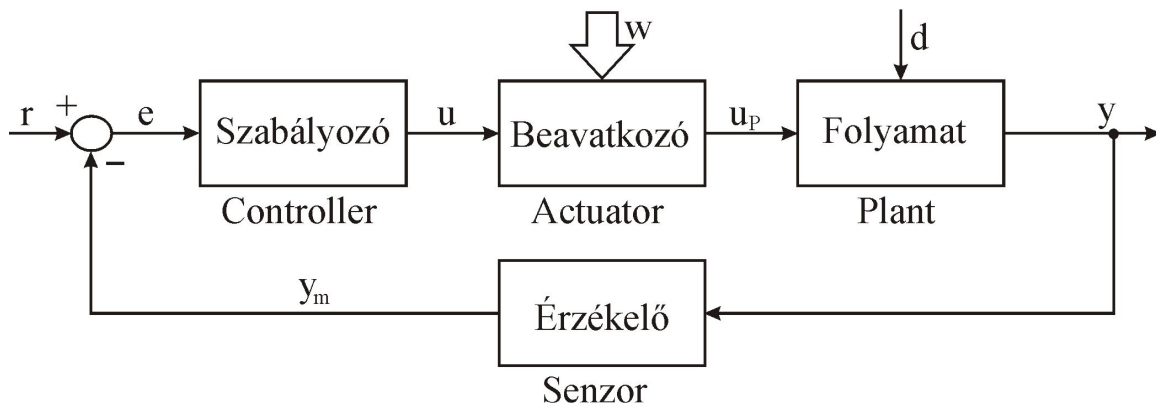
1. A gyakorlat célja

A hőmérsékletszabályozási hurok elemeinek megismerése, a folyamat számítógép interfész megvalósítása. Kapcsoló üzemmódú (ON-OFF) szabályozó megvalósítása és tanulmányozása.

2. Elméleti bevezető

2.1 . A szabályozási hurok

A szabályozási hurok a folyamatból és az annak az irányítását biztosító technikai eszközökből álló zárt rendszer (lásd 1. Ábra).



1. Ábra: A szabályozási hurok felépítése

A hurokban megjelenő jelek elnevezései:

u_p – a folyamat bemenete – plant input (módosított jellemző)

y – a folyamat kimenete – plant output (szabályozott jellemző)

y_m – mért kimenet – measured output (ellenőrző jel)

r – előírt érték – reference signal (alapjel)

e – szabályozási hiba – control error (rendelkező jel)

u – beavatkozó jel – control signal

d – zavaró jel – disturbance

A szabályozási hurok elemeinek főbb jellemzői az irányítási algoritmus szempontjából:

Irányított folyamat: Bemenete tipikusan nagyenergiájú jel, amelyet a szabályozási hurokban belül úgy kell meghatározni, hogy a folyamat kimenete kövesse az előírt értéket.

Érzékelő: a folyamat kimenetét alakítja az irányítási algoritmus által feldolgozható értékre. A bemeneteük lehet nem-elektromos jel (például hőmérséklet, fordulatszám), a kimenete tipikusan elektromos jel. Célszerű, ha az érzékelő kimenete egyezményesített,

szabványos tartományú jel. Ilyen tartományok például a 0-5 V, 0-10 V, 0-20 mA, 4-20 mA. Természetesen nem minden jelet lehet vagy érdemes egyezményesített tartományba átalakítani.

Beavatkozó: A beavatkozó szerepe, hogy bemenetén megjelenő kisenergiájú jelet (amit a szabályozó határoz meg) a folyamat bemenetének megfelelő szintre átalakítsa, tehát tipikusan kisenergiájú jelből nagyenergiájú jelet kell, hogy létrehozzon. Jellemző a beavatkozóra, hogy ezen az elemen keresztül áramlik be az energia a szabályozási körbe. Hasznos, ha a bemenete egyezményesített jel. Elvárható, hogy kisebb időállandókkal rendelkezzen, mint az irányított folyamat.

Az *irányítási algoritmus* feladata, hogy az érzékelő által mért kimenet és az előírt érték függvényében meghatározza a beavatkozó bemenetét úgy, hogy a folyamat kimenete kövesse az alapjelet. Az esetek többségében először képezzük az előírt érték és a mért kimenet különbségét, vagyis meghatározzuk a szabályozási hibát:

$$e = r - y_m \quad (1)$$

A szabályozó bemenetét így a szabályozási hiba képezi. A szabályozónak mind a bemenete mind a kimenete kisenergiájú jel.

2.2 . Mintavételes szabályozások megvalósítása

A mintavételes kialakítás felvet néhány speciális problémát: az analóg-digitális, digitális-analóg átalakítás miatti kerekítési hibák, a mért jelek digitális szűrése, kalibrációja, a mintavételi periódus meghatározása.

Analóg-digitális, digitális analóg átalakítás: Az analóg digitális átalakítás miatt kvantálási hibára számíthatunk a mért jel beolvasásánál, a mérés pontossága az átalakító felbontásától függ. Ha az átalakító felbontása N , a mért jel y_{MIN} , y_{MAX} tartományban van, akkor az elérhető mérési pontosság $(y_{MIN} - y_{MAX})/2^N$, ha a mérés lineáris. Ezért az analóg-digitális átalakítót úgy kell megválasztani, hogy nagyobb pontossággal tudjunk mérni, mint az elvárt szabályozási pontosság. A digitális-analóg átalakító (a beavatkozó jel kiküldése) felbontásának meghatározásánál ugyancsak szabályozási pontosságot kell figyelembe venni. Ha nagy felbontással tudjuk csak módosítani a beavatkozó jelet, a folyamat kimenete is nagy felbontással fog változni. A napjainkban ipari alkalmazásokra elterjedt analóg-digitális, digitális-analóg konverterek felbontása tipikusan 10-16 bit között van, de ennél nagyobb felbontású átalakítók is kaphatóak elérhető áron.

Mérések kalibrálása: Az analóg-digitális átalakítóról beolvasott érték egy egész szám $0 \dots 2^N$ tartományban, ahol N az átalakító pontossága. A beolvasott jelet mindig át kell alakítani az irányítási algoritmus által feldolgozható, valamint a kezelőszemélyzet által értelmezhető értékre. Kalibrálás alatt az analóg-digitális átalakítóról beolvasott érték transzformálását értjük szabványos (pl. SI) mértékegységekkel jellemezhető tartományba. Amennyiben a mérési tartományban az érték lineárisan változik, a kalibráció

$y_M = a_M y_{AD} + b_M$ lineáris transzformációt jelent. Ha a mérés tartománya $y_{MIN} \dots y_{MAX}$ és az átalakító pontossága N (y_{MIN} bemenetre 0 -t, y_{MAX} bemenetre 2^N -et olvasunk be), akkor:

$$\begin{aligned}
y_M &= a_M y_{AD} + b_M \\
a_M &= \frac{y_{MAX} - y_{MIN}}{2^N} \\
b_M &= y_{MIN}
\end{aligned}
\tag{2}$$

Amennyiben a mérés nem változik lineárisan a mérési tartományban, a kalibrációhoz nem elsőfokú, hanem magasabb fokú polinomot alkalmazhatunk. A kalibráció előtt a beolvasott érték még egész előjel nélküli szám, a kalibráció után már célszerű lebegőpontos ábrázolást alkalmazni.

A jelek szűrése: Az ipari mérések esetén mindig kell számolni mérési zajokkal. A mérési zajok miatt az irányítási algoritmus bemenetén ‘torzítva’ jelenik meg a folyamat kimenete, ezért az irányítás minősége romolhat.

A mérési zajok általában nagy frekvencián jelentkeznek, ezért az alkalmazott szűrő aluláteresztő szűrő kell, hogy legyen. A szűrő megválasztásnál becslésünk kell legyen arról, hogy a mérési zajok milyen frekvenciatartományban jelentkeznek. Irányítástechnikai feladatokban általában elégséges egyszerű struktúrájú, kis foksámú szűrőket alkalmazni.

Lassú irányítások (például hőmérsékletszabályozás) esetében szóba jöhetnek véges impulzusválaszú (FIR – Finite Impulse Response) *átlagoló szűrők*. Ebben az esetben a szűrő kimenete az utolsó N mérés átlaga:

$$y_{Fk} = \sum_{i=0}^{N-1} \frac{y_{k-i}}{N}
\tag{3}$$

y_{Fk} a szűrő kimenete, y_k a mért érték a k -ik mintavételben. Az N értékével állíthatjuk be. Minél alacsonyabb frekvencián jelentkeznek a zajok, annál nagyobb N -t szükséges választani.

Gyors irányításoknál (például pozíciószabályozásnál az inkrementális adóról beolvasott frekvenciaérték szűrése) alkalmazhatunk *rekurzív szűrőket*. Sok esetben elégséges egy elsőfokú rendszer mintavételes megvalósítása. Közismert, hogy a stabil elsőfokú rendszer aluláteresztő szűrő.

$$H_F(s) = \frac{Y_F(s)}{Y(s)} = \frac{1}{T_F s + 1}
\tag{4}$$

T_F a rendszer időállandója. A szűrő erősítése egységnyinek választható. Minél alacsonyabb frekvencián jelentkeznek a mérési zajok, annál nagyobbra kell választani T_F értékét. A szűrő mintavételes megvalósításánál kiindulhatunk az őt leíró differenciálegyenlet alakjából, amire a hátrtartó differenciák közelítést alkalmazunk, lásd (5). T a mintavételi periódust jelöli. A szűrő kimenete az előző mintavételbeli szűrt értéktől is függ.

$$T_F \frac{dy_F}{dt} + y_F = y$$

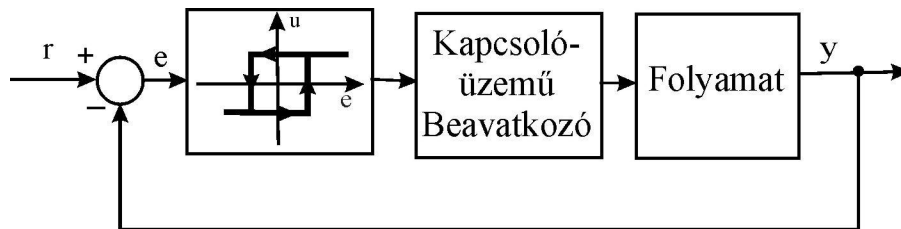
$$T_F \frac{y_{Fk} - y_{Fk-1}}{T} + y_{Fk} = y_k \quad (5)$$

$$y_{Fk} = \frac{T_F}{T_F + T} \cdot y_{Fk-1} + \frac{T}{T_F + T} \cdot y_k$$

2.3 Kétállású (ON-OFF) szabályozó

Számos irányítástechnikai alkalmazásnál célszerű kapcsoló üzemmódú szabályozót alkalmazni, vagy a szabályozót kibővíteni nemfolytonos elemekkel.

A kétállású szabályozó a hiba függvényében két fix értéket képes kiadni ($u=+1; u=-1$ vagy $u=+1; u=0$), ennek megfelelően a beavatkozó szerv is két üzemmódban működhet a kiadott beavatkozó jel függvényében. A túl gyors kapcsolás elkerülésére a szabályozót (h szélességű) hiszterézissel módosíthatjuk. (lásd 10.24. Ábra)



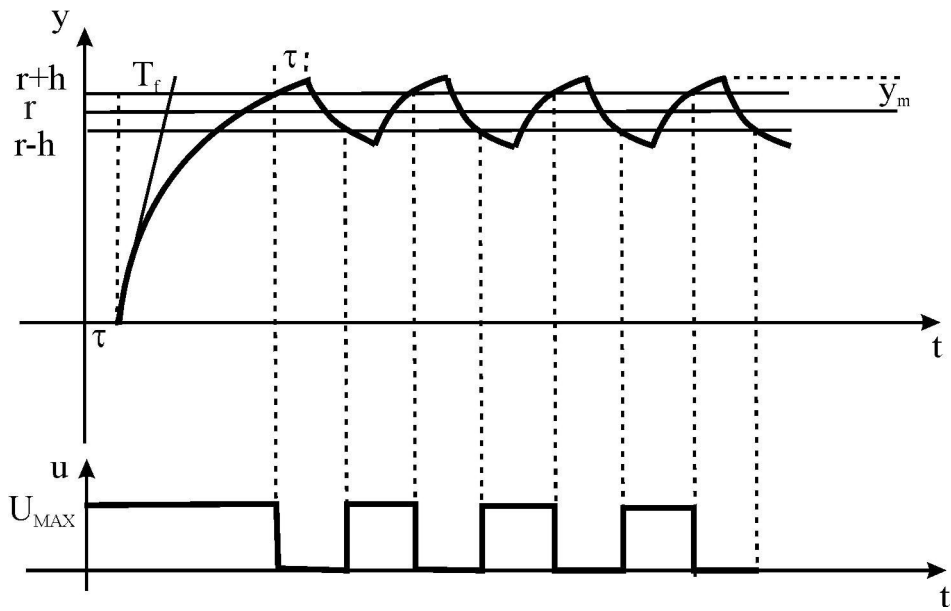
2. Ábra: Szabályozási rendszer kétállású szabályozóval

Vizsgáljuk meg azt az esetet is, amikor az irányított folyamat *elsőfokú átviteli függvény*vel modellezhető és a rendszer *tartalmaz holtidőt*.

$$H_f(s) = \frac{K_f}{T_f s + 1} e^{-s\tau} \quad (6)$$

τ a holtidőt jelöli.

Az irányított folyamat válasza hasonló lesz az előző esethez, de a holtidő miatt a lengések periódusa megnő és a maximális eltérés az előírt értéktől ugyancsak nagyobb lesz (lásd 3 Ábra). Az irányítási rendszer válaszában tulajdonságait a hiszterézis szélessége valamint a rendszer időállandói nagymértékben meghatározzák. A hiszterézis szélessége a legnagyobb eltérésre és a lengések tartományára is hatással van. A hiszterézis szélességének csökkentése nagyobb kapcsolási frekvenciát von maga után.



3 Ábra: Kétállású szabályozóval irányított holtidős rendszer válasza

3. A mérés menete

Az irányítás célja egy félmapka hőmérsékletének konstans értékre szabályozása. Tehát az irányított folyamat bemenete a félmapkának leadott hőmennyiség, kimenete a félmapka hőmérséklete.

A félmapka hőmérsékletét egy LM135 típusú kalibrált félvezető alapú hőmérsékletsenzorral mérjük (érzékelő). Az érzékelő $-40 \dots +135$ Celsius fokos hőmérséklettartományban alkalmazható és kisebb mint 1 Celsius fok a mérési pontossága.

A működésének feltétele, hogy a rajta átfolyó áram $0.5 \dots 5$ milliAmper tartományban legyen. Ezt úgy oldhatjuk meg, hogy az egyenfeszültséggel történő betápláláskor az érzékelővel párhuzamosan egy ellenállást helyezünk el. Ha a tápfeszültség U_T és az érzékelőn eső feszültség $U_E = 0.5 \text{ mV}$, akkor az ellenállás értékét $R = (U_T - U_E) / I$ alapján kell megválasztani, ahol I az érzékelőn átfolyó (elvárt) áram.

Az érzékelő Kelvinben van kalibrálva. 0 Kelvin foknak 0 V felel, meg és ha a hőmérséklet 1 fokot változik, akkor az érzékelőn mért feszültség 10 millivoltot változik. Így ha a hőmérséklet 0 Celsius fok, az érzékelőn $2,73$ Voltot mérünk.

A félmapkának a hőt egy kapcsolóüzemű teljesítménytranzisztorral adjuk le. Ugyanakkor a félmap hűtésére egy ventilátort alkalmazunk. Tehát a beavatkozó a tranzisztorból és a ventilátorból álló együttes.

A számítógéphez a rendszert egy OMEGA DAS 08-as adatbegyűjtő kártya segítségével csatoljuk. Az érzékelő által adott feszültséget az adatbegyűjtő kártya egyik analóg bemenetére csatoljuk. A tranzisztor és a ventilátort a kártya egy-egy digitális kimenetére csatoljuk.

Elvégzendő programozási feladatok:

I. A kártya meghajtójának illesztése a programunkhoz:

A kártya funkcióit C++ könyvtárfüggvényeken keresztül érhetjük el. Azt a *TempControlLab* tervhez (a program interfészét lásd a 4 Ábrán) az *Add Files to Project* menüpontban illeszthetjük. Ahhoz, hogy a *TemperatureControl.cpp* állományban a kártyafüggvényeket alkalmazni tudjuk, az állományba be kell illeszteni a *cbw.h* interfészt.

II. Hőmérséklet beolvasása: *ReadADConversionResult* függvényben. A visszatérített érték a beolvasott AD konverzió eredménye. Ehhez az alábbi függvényt alkalmazzuk:

cbAIIn(BOARD_NUM, ANALOG_CHANNEL, ADRANGE, &ADConversionResult)

BOARD_NUM – az adatbegyűjtő kártyához a meghajtó által hozzárendel szám (jelen esetben 1)

ANALOG_CHANNEL – az analóg csatorna száma, amire a beolvasott értéket kötöttük (jelen esetben 0)

ADRANGE – a mérési tartomány, amelyben beolvassuk a jeleket (jelen esetben +/- 5V, az ennek megfelelő konstans BIP5VOLTS)

ADConversionResult – *unsigned short* típusú változó amibe az konverzió eredménye kerül.

III. A hőmérséklet kalibrálása és szűrése: *CalibrateTemperature* függvényben. A visszatérített érték a szűrt hőmérsékletérték.

A hőmérséklet kalibrálásához először átalakítjuk a beolvasott értéket voltba. A kártyán az AD konverzió 12 bites, tehát a beolvasott értékek tartománya 0 ... 4096. A mérési tartomány -5V ... +5V, tehát a feszültséget az alábbi módon számíthatjuk:

$$U = (5 - (-5)) / 4096 * (ADConversionResult - 2048)$$

A második értékben a kiszámított feszültséget átalakítjuk hőmérsékletté (Celsius fokba). Mivel 0 fok Celsiusra a feszültség 2.73 V és 1 fok hőmérsékletváltozásra a feszültség 10 mV-ot változik:

$$T = 100 * U - 273$$

A harmadik lépésben a kapott hőmérsékletértéket szűrjük. Ehhez a (3) összefüggést alkalmazzuk $N=3$ ra. A T_{k-1} , T_{k-2} értékeket statikus változóként kell elmenteni. És értékeket a szűrt hőmérséklet számolása után frissíteni kell:

$$T_{k-2} = T_{k-1}$$

$$T_{k-1} = T_k$$

III. A szabályozási hiba számolása: *CalculateControlError* függvényben. A visszatérített érték a szabályozási hiba, amit az (1) összefüggés alapján számolunk. Hogy számolni tudjuk a függvényben meg kell hívni a *GetControlReference* és a *CalibrateTemperature* függvényeket.

IV. A beavatkozó jel számolása és kiküldése: *CalculateControlSignal* függvényben. Az ON-OFF szabályozás elve alapján, ha a szabályozási hiba előjele pozitív, akkor a beavatkozó jel értéke -1 (OFF, vagyis a ventilátor van bekapcsolva), máskülönben a beavatkozó jel értéke +1 (ON, vagyis a fűtő tranzisztor van bekapcsolva). A digitális beavatkozó jelek kiküldéséhez alkalmazzuk az alábbi függvényt:

cbDOut(BOARD_NUM, PORT,value)

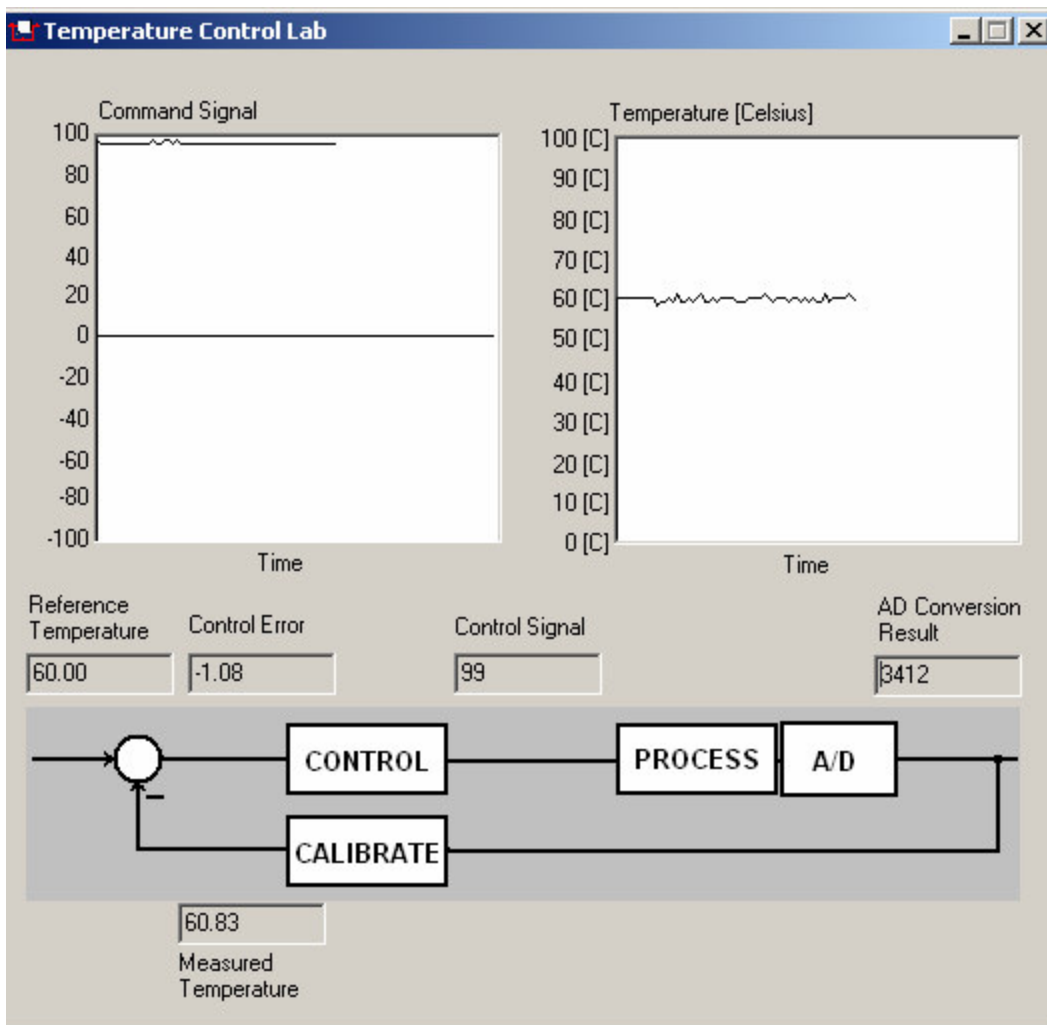
BOARD_NUM – az adatbegyűjtő kártyához a meghajtó által hozzárendelt szám (jelen esetben 1)

PORT – a digitális port, ahova a jeleket szeretném kiküldeni (jelen esetben a port neve AUXPORT).

value – a digitális portra kiküldendő érték

Mivel a fűtő tranzisztor a nulladik bitjére, a hűtést biztosító ventilátor a port első bitjére van kötve, ezért ON esetén 0X01-et, OFF esetén 0X02-t kell kiküldeni a portra.

A szabályozási rendszer jeleinek numerikus kiírása automatikusan történik. A grafikus kiíratáshoz alkalmazzuk a *GenerateTemperatureList* függvényt, módosítva a hőmérsékletkiíratásnak megfelelően.



4. Ábra. A program interfésze

4. Kérdések és feladatok

1. A grafikusan és numerikusan kijelzett hőmérséklet alapján állapítsuk meg, hogy mennyi a rendszer holtideje.
2. A grafikusan és numerikusan kijelzett mérések alapján állapítsuk meg, hogy mennyi a hőmérséklet jel lengések amplitúdója.
3. Módosítsuk a szabályozást úgy, hogy a szabályozóban helyezzünk el hiszterézis elemet is. A hiszterézis szélessége legyen 2 fok. Vizsgáljuk meg, hogyan módosul a szabályozási rendszer hőmérsékletválasza.